CLAIMS

1. A method for performing path-sensitive value flow analysis on an abstract program, the abstract program having a plurality of statements that are derived from complex statements coded within a software program, the method comprising:

tracking a concrete state and value alias information for each statement along each relevant path in the abstract program;

storing the concrete state and value alias information for each statement along each relevant path in a symbolic store, the symbolic store storing a plurality of symbolic states, each symbolic state comprising the concrete state at a specific location along a specific relevant path in the abstract program and the value alias information at the specific location along the specific relevant path, the set of aliases being associated with a designated value that is being analyzed;

upon encountering a decisional statement, proceeding individually along each decision path associated with the decisional statement as long as the concrete state in the symbolic state being processed at the decisional statement reflects that the decision path is relevant; and

merging two symbolic states in the symbolic store if the two symbolic states exist for the same specific location in the abstract program and if the value alias information in the two symbolic states are identical.

2. The method of claim 1, wherein the two symbolic states are merged by deleting information in the concrete state of both symbolic states if the information differs between the two symbolic states.

3. The method of claim 1, wherein the value alias information is obtained using imprecise memory alias analysis.

4. The method of claim 1, wherein the value alias information includes a first set of aliases that identify aliases for the designated value and a second set of aliases that identify possible aliases for the designated value.

5. The method of claim 4, wherein the second set of aliases is over-inclusive.

6. The method of claim 4, wherein the first set and second set of aliases are determined by performing transform functions on the first and second set of aliases based on a type of statement that is being processed in the abstract program.

7. The method of claim 6, wherein the type of statement includes an initial statement, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the first set of aliases, the remove transfer function is the empty set.

8. The method of claim 6, wherein the type of statement includes a scalar assignment in which a variable Y is assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the

first set if the variable Y was in the first set before the statement and adds a field pointed to by variable X to the first set if the field derefenced by variable Y was in the first set before the statement, the remove transfer function removes fields referenced by the variable X, and removes the variable X and any field that points to the same memory location as variable X, if the variable Y was not in the first set before the statement.

9. The method of claim 6, wherein the type of statement includes an assignment in which an address of variable Y is assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the dereference of variable X to the first set if the variable Y was in the first set before the statement, the remove transfer function removes fields referenced with the variable X, removes the variable X, and removes any field that points to the same memory location as variable X.

10. The method of claim 6, wherein the type of statement includes a call to a memory allocation function with a return value assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function is an empty set, the remove transfer function removes fields referenced with the variable X, removes the variable X, and removes any field that points to the same memory location as variable X.

11. The method of claim 6, wherein the type of statement includes an assignment in which a field F pointed to by variable Y is assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the first set if the field F pointed to by the variable Y was in the first set before the statement, the remove transfer function removes fields referenced by the variable X, and removes the variable X and any field that points to the same memory location as variable X, if the variable Y was not in the first set before the statement.

12. The method of claim 6, wherein the type of statement includes an assignment in which an address of a field F referenced by variable Y is assigned to variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the dereference of variable X to the first set if the field F referenced by variable Y was in the first set before the statement, the remove transfer function removes fields referenced with the variable X, removes the variable X, and removes any field that points to the same memory location as variable X.

13. The method of claim 6, wherein the type of statement includes an assignment in which a variable Y is assigned to a field F referenced by variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the field F referenced by variable X if the variable Y was in the first set before the statement, the remove transfer function removes any field referenced by a variable

Z such that the variable Z refers to the same memory location as the field F referenced by the variable X, and if the variable Y was not in the first set, removes the field F referenced by the variable X, any field that points to the same memory location as the field F referenced by the variable X, and any variable that refers to the same memory location as the field F referenced by the variable X before the statement.

14. The method of claim 6, wherein the transform functions for the type of statement include a generate transfer function and a remove transfer function for determining the first set of aliases, the generate transfer function is the empty set, the remove transfer function removes any variable that pointed to the same memory location as a memory cell that was updated by the statement and removes any field if the memory cell updated by the statement pointed to the same memory location as the field or the pointer.

15. The method of claim 6, wherein the type of statement includes an initial statement, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the second set of aliases, the remove transfer function is the empty set.

16. The method of claim 6, wherein the type of statement includes a scalar assignment in which a variable Y is assigned to a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the

second set if the variable Y refers to the same memory location as one of the aliases in the second set of aliases before the statement, the remove transfer function removes the variable X.

17. The method of claim 6, wherein the type of statement includes an assignment in which an address of variable Y is assigned to a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function is the empty set, the remove transfer function removes the variable X from the second set.

18. The method of claim 6, wherein the type of statement includes a call to a memory allocation function with a return value assigned to a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function is the empty set, the remove transfer function removes the variable X from the second set.

19. The method of claim 6, wherein the type of statement includes an assignment in which a field F pointed to by variable Y is assigned to a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X if the field F pointed to by variable Y refers to the same memory location as one of the aliases in the second set of aliases before the statement, the remove transfer function removes the variable X from the second set of aliases.

20. The method of claim 6, wherein the type of statement includes an assignment in which an address of a field F referenced by variable Y is assigned to variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function is the empty set, the remove transfer function removes the variable X from the second set.

21. The method of claim 6, wherein the type of statement includes an assignment in which a variable Y is assigned to a field F referenced by variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the field F referenced by variable X to the second set if the variable Y refers to the same memory location as one of the aliases in the second set before the statement, the remove transfer function is the empty set.

22. The method of claim 6, wherein the transform functions for the type of statement include a generate transfer function and a remove transfer function for determining the second set of aliases, the generate transfer function adds an expression if the expression was updated by the statement or if one of the memory locations looked up when executing the statement pointed to a memory location in the second set of aliases, the remove transfer function is the empty set.

23. A computer-readable medium for performing path-sensitive value flow analysis, comprising:

applying transform functions to determine value alias information based on a type of statement that is being processed in an abstract program, the value alias information comprising a first set of aliases that identify aliases for a designated value that is being analyzed and a second set of aliases that identify possible aliases for the designated value, a portion of the value alias information being obtained using imprecise memory alias analysis.

24. The computer-readable medium of claim 23, wherein the type of statement includes an initial statement, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the first set of aliases, the remove transfer function is the empty set.

25. The computer-readable medium of claim 23, wherein the type of statement includes a scalar assignment in which a variable Y is assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the first set if the variable Y was in the first set before the statement and adds a field pointed to by variable X to the first set if the field derefenced by variable Y was in the first set before the statement, the remove transfer function removes fields referenced by the variable X, and removes the variable X and any field that points to the same memory location as variable X, if the variable Y was not in the first set before the statement.

26. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which an address of variable Y is assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the dereference of variable X to the first set if the variable Y was in the first set before the statement, the remove transfer function removes fields referenced with the variable X, removes the variable X, and removes any field that points to the same memory location as variable X.

27. The computer-readable medium of claim 23, wherein the type of statement includes a call to a memory allocation function with a return value assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function is an empty set, the remove transfer function removes fields referenced with the variable X, removes the variable X, and removes any field that points to the same memory location as variable X.

28. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which a field F pointed to by variable Y is assigned to a variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the first set if the field F pointed to by the variable Y was in the first set before the statement, the remove transfer function removes fields referenced by the variable X, and removes the variable X and any field that

points to the same memory location as variable X, if the variable Y was not in the first set before the statement.

29. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which an address of a field F referenced by variable Y is assigned to variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the dereference of variable X to the first set if the field F referenced by variable Y was in the first set before the statement, the remove transfer function removes fields referenced with the variable X, removes the variable X, and removes any field that points to the same memory location as variable X.

30. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which a variable Y is assigned to a field F referenced by variable X, the transform functions for the first set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the field F referenced by variable X if the variable Y was in the first set before the statement, the remove transfer function removes any field referenced by a variable Z such that the variable Z refers to the same memory location as the field F referenced by the variable X, and if the variable Y was not in the first set, removes the field F referenced by the variable X, any field that points to the same memory location as the field F referenced by the variable X, and any variable that refers to the same memory location as the field F referenced by the variable X before the statement.

31. The computer-readable medium of claim 23, wherein the transform functions for the type of statement include a generate transfer function and a remove transfer function for determining the first set of aliases, the generate transfer function is the empty set, the remove transfer function removes any variable that pointed to the same memory location as a memory cell that was updated by the statement and removes any field if the memory cell updated by the statement pointed to the same memory location as the field or the pointer.

32. The computer-readable medium of claim 23, wherein the type of statement includes an initial statement, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the second set of aliases, the remove transfer function is the empty set.

33. The computer-readable medium of claim 23, wherein the type of statement includes a scalar assignment in which a variable Y is assigned to a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X to the second set if the variable Y refers to the same memory location as one of the aliases in the second set of aliases before the statement, the remove transfer function removes the variable X.

34. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which an address of variable Y is assigned to

a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function is the empty set, the remove transfer function removes the variable X from the second set.

35. The computer-readable medium of claim 23, wherein the type of statement includes a call to a memory allocation function with a return value assigned to a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function is the empty set, the remove transfer function removes the variable X from the second set.

36. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which a field F pointed to by variable Y is assigned to a variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the variable X if the field F pointed to by variable Y refers to the same memory location as one of the aliases in the second set of aliases before the statement, the remove transfer function removes the variable X from the second set of aliases.

37. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which an address of a field F referenced by variable Y is assigned to variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the

generate transfer function is the empty set, the remove transfer function removes the variable X from the second set.

38. The computer-readable medium of claim 23, wherein the type of statement includes an assignment in which a variable Y is assigned to a field F referenced by variable X, the transform functions for the second set of aliases include a generate transfer function and a remove transfer function, the generate transfer function adds the field F referenced by variable X to the second set if the variable Y refers to the same memory location as one of the aliases in the second set before the statement, the remove transfer function is the empty set.

39. The computer-readable medium of claim 23, wherein the transform functions for the type of statement include a generate transfer function and a remove transfer function for determining the second set of aliases, the generate transfer function adds an expression if the expression was updated by the statement or if one of the memory locations looked up when executing the statement pointed to a memory location in the second set of aliases, the remove transfer function is the empty set.

40. A system for performing path-sensitive value flow analysis on an abstract program, the abstract program having a plurality of statements that were derived from complex statements coded within a software program, the system comprising:

a processor; and

a memory into which a plurality of instructions are loaded, the plurality of instructions performing a method comprising:

tracking a concrete state and value alias information for each statement along each relevant path in the abstract program;

storing the concrete state and value alias information for each statement along each relevant path in a symbolic store, the symbolic store storing a plurality of symbolic states, each symbolic state comprising the concrete state at a specific location along a specific relevant path in the abstract program, the value alias information at the specific location along the specific relevant path, the set of aliases being associated with a designated value that is being analyzed;

upon encountering a decisional statement, proceeding individually along each decision path associated with the decisional statement as long as the concrete state in the symbolic state at the decisional statement reflects that the decision path is relevant; and

merging two symbolic states in the symbolic store if the two symbolic states exist for the same specific location in the abstract program and if the value alias information in the two symbolic states are identical.

41. The system of claim 40, wherein the two symbolic states are merged by deleting information in the concrete state of both symbolic states if the information differs between the two symbolic states.

42. The system of claim 40, wherein the value alias information is obtained using imprecise memory alias analysis.

43. The system of claim 40, wherein the value alias information includes a first set of aliases that identify aliases for the designated value and a second set of aliases that identify possible aliases for the designated value.

44. The system of claim 43, wherein the second set of aliases is over-inclusive.

45. The system of claim 43, wherein the first set and second set of aliases are determined by performing transform functions on the first and second set of aliases based on a type of statement that is being processed in the abstract program.